Attorney Docket No. 0026-0008

UNITED STATES PATENT APPLICATION

OF

Alexander Mark FRANZ, Monika H. HENZINGER,
Sergey BRIN, and Brian Christopher MILCH

FOR

VOICE INTERFACE FOR A SEARCH ENGINE

[0001] VOICE INTERFACE FOR A SEARCH ENGINE

[0002] FIELD OF THE INVENTION

[0003] The present invention relates generally to information retrieval systems and, more particularly, to a system and method for supporting voice queries in information retrieval systems.

[0004] BACKGROUND OF THE INVENTION

[0005] To satisfy the average user, a voice interface to a search engine must recognize spoken queries, and must return highly relevant search results. Several problems exist in designing satisfactory voice interfaces. Current speech recognition technology has high word error rates for large vocabulary sizes. There is very little repetition in queries, providing little information that could be used to guide the speech recognizer. In other speech recognition applications, the recognizer can use context, such as a dialogue history, to set up certain expectations and guide the recognition. Voice search queries lack such context. Voice queries can be very short (on the order of only a few words or single word), so there is very little information in the utterance itself upon which to make a voice recognition determination.

[0006] Current voice interfaces to search engines address the above problems by limiting the scope of the voice queries to a very narrow range. At every turn, the user is prompted to select from a small number of choices. For example, at the initial menu, the user might be able to choose from "news," "stocks," "weather," or "sports." After the user chooses one category, the system offers another small set of choices. By limiting the

number of possible utterances at every turn, the difficulty of the speech recognition task is reduced to a level where high accuracy can be achieved. This approach results in an interactive voice system that has a number of severe deficiencies. It is slow to use, since the user must navigate through may levels of voice menus. If the user's information need does not match a predefined category, then it becomes very difficult or impossible to find the information desired. Moreover, it is often frustrating to use, since the user must adapt his/her interactions to the rigid, mechanical structure of the system.

[0007] Therefore, there exists a need for a voice interface that is effective for search engines.

[0008] SUMMARY OF THE INVENTION

[0009] A system and method consistent with the present invention address this and other needs by providing a voice interface for search engines that is capable of returning highly relevant results.

[0010] In accordance with the purpose of the invention as embodied and broadly described herein, a method that provides search results includes receiving a voice search query from a user; deriving one or more recognition hypotheses from the voice search query, each recognition hypothesis being associated with a weight; constructing a weighted boolean query using the recognition hypotheses; providing the weighted boolean query to a search system; and providing results of the search system.

[0011] In another implementation consistent with the present invention, a server includes a memory and a processor. The processor receives one or more recognition

hypotheses. The recognition hypotheses are constructed from a voice search query. The processor also determines the length of the shortest recognition hypothesis, prunes the length of each recognition hypothesis up to the length of the shortest recognition hypothesis, determines a length of a longest pruned recognition hypothesis, selects a number of recognition hypotheses based on a value representing the length of the longest recognition hypothesis, determines query term weights, and forms a weighted boolean query out of each word position in the selected recognition hypotheses.

[0012] BRIEF DESCRIPTION OF THE DRAWINGS

[0013] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate an embodiment of the invention and, together with the description, explain the invention. In the drawings,

[0014] FIG. 1 illustrates an exemplary network in which a system and method, consistent with the present invention, may be implemented;

[0015] FIG. 2 illustrates an exemplary client device consistent with the present invention;

[0016] FIG. 3 illustrates an exemplary server consistent with the present invention;

[0017] FIG. 4 illustrates an exemplary process, consistent with the present invention, for producing models for use in voice-based searching;

[0018] FIG. 5 illustrates an exemplary process, consistent with the present invention, for performing a search;

[0019] FIGS. 6A and 6B illustrate an exemplary n-best hypothesis list and a word graph, respectively, consistent with the present invention; and

[0020] FIG. 7 illustrates an exemplary process, consistent with the present invention, for constructing a search query.

[0021] DETAILED DESCRIPTION

[0022] The following detailed description of the invention refers to the accompanying drawings. The same reference numbers in different drawings identify the same or similar elements. Also, the following detailed description does not limit the invention. Instead, the scope of the invention is defined by the appended claims.

[0023] Implementations consistent with the present invention provide a voice interface to search engines. In response to a voice query, a server automatically constructs a search query to cover the most likely hypotheses identified by a speech recognizer.

[0024] EXEMPLARY NETWORK

[0025] FIG. 1 illustrates an exemplary network 100 in which a system and method, consistent with the present invention, may be implemented. The network 100 may include multiple client devices 110 connected to multiple servers 120-130 via a network 140. The network 140 may include a local area network (LAN), a wide area network (WAN), a telephone network, such as the Public Switched Telephone Network (PSTN), an intranet, the Internet, or a combination of networks. Two client devices 110 and three servers 120-130 have been illustrated as connected to network 140 for simplicity. In

practice, there may be more or less client devices and servers. Also, in some instances, a client device may perform the functions of a server and a server may perform the functions of a client device.

[0026] The client devices 110 may include devices, such as mainframes, minicomputers, personal computers, laptops, personal digital assistants, telephones, or the like, capable of connecting to the network 140. The client devices 110 may transmit data over the network 140 or receive data from the network 140 via a wired, wireless, or optical connection.

[0027] The servers 120-130 may include one or more types of computer systems, such as a mainframe, minicomputer, or personal computer, capable of connecting to the network 140 to enable servers 120-130 to communicate with the client devices 110. In alternative implementations, the servers 120-130 may include mechanisms for directly connecting to one or more client devices 110. The servers 120-130 may transmit data over network 140 or receive data from the network 140 via a wired, wireless, or optical connection.

[0028] In an implementation consistent with the present invention, the server 120 may include a search engine 125 usable by the client devices 110. The servers 130 may store documents, such as web pages, accessible by the client devices 110.

[0029] EXEMPLARY CLIENT ARCHITECTURE

[0030] FIG. 2 illustrates an exemplary client device 110 consistent with the present invention. The client device 110 may include a bus 210, a processor 220, a main memory

230, a read only memory (ROM) 240, a storage device 250, an input device 260, an output device 270, and a communication interface 280. The bus 210 may include one or more conventional buses that permit communication among the components of the client device 110.

[0031] The processor 220 may include any type of conventional processor or microprocessor that interprets and executes instructions. The main memory 230 may include a random access memory (RAM) or another type of dynamic storage device that stores information and instructions for execution by the processor 220. The ROM 240 may include a conventional ROM device or another type of static storage device that stores static information and instructions for use by the processor 220. The storage device 250 may include a magnetic and/or optical recording medium and its corresponding drive.

[0032] The input device 260 may include one or more conventional mechanisms that permit a user to input information to the client device 110, such as a keyboard, a mouse, a pen, a microphone, voice recognition and/or biometric mechanisms, etc. The output device 270 may include one or more conventional mechanisms that output information to the user, including a display, a printer, a speaker, etc. The communication interface 280 may include any transceiver-like mechanism that enables the client device 110 to communicate with other devices and/or systems. For example, the communication interface 280 may include mechanisms for communicating with another device or system via a network, such as network 140.

[0033] As will be described in detail below, the client devices 110, consistent with the present invention, perform certain searching-related operations. The client devices 110 may perform these operations in response to processor 220 executing software instructions contained in a computer-readable medium, such as memory 230. A computer-readable medium may be defined as one or more memory devices and/or carrier waves.

[0034] The software instructions may be read into memory 230 from another computer-readable medium, such as the data storage device 250, or from another device via the communication interface 280. The software instructions contained in memory 230 causes processor 220 to perform the search-related activities described below. Alternatively, hardwired circuitry may be used in place of or in combination with software instructions to implement processes consistent with the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

[0035] EXEMPLARY SERVER

[0036] FIG. 3 illustrates an exemplary server 120 consistent with the present invention. Server 130 may be similarly configured. The server 120 includes a bus 310, a processor 320, a memory 330, an input device 340, an output device 350, and a communication interface 360. The bus 310 may include one or more conventional buses that allow communication among the components of the server 120.

[0037] The processor 320 may include any type of conventional processor or microprocessor that interprets and executes instructions. The memory 330 may include a RAM or another type of dynamic storage device that stores information and instructions for execution by the processor 320; a ROM or another type of static storage device that stores static information and instructions for use by the processor 320; and/or some type of magnetic or optical recording medium and its corresponding drive.

[0038] The input device 340 may include one or more conventional devices that permits an operator to input information to the server 120, such as a keyboard, a mouse, a pen, a microphone, voice recognition and/or biometric mechanisms, and the like. The output device 350 may include one or more conventional devices that outputs information to the operator, including a display, a printer, a speaker, etc. The communication interface 360 may include any transceiver-like mechanism that enables the server 120 to communicate with other devices and/or systems. For example, the communication interface 360 may include mechanisms for communicating with other servers 130 or the client devices 110 via a network, such as network 140.

[0039] Execution of the sequences of instructions contained in memory 330 causes processor 320 to perform the functions described below. In alternative embodiments, hardwired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software.

[0040] EXEMPLARY PROCESSING

[0041] FIG. 4 illustrates an exemplary process, consistent with the present invention, for producing models for use in voice-based searching. In an implementation consistent with the present invention, a server, such as server 120, may perform this process. It will be appreciated, however, that a client device 110 may alternatively perform the entire process or part of the process described below.

[0042] Processing may begin with the server 120 receiving search query logs (i.e., one or more previously executed queries) [act 405]. The query logs may consist of audio data (i.e., a recorded query) and/or a textual transcription of the audio data. The textual transcription may be obtained manually or, as will be described in more detail below, may be automatically performed by the server 120. The query logs may also consist of typed query logs from, for example, a text-based search engine.

[0043] The server 120 may filter the query log to remove unwanted data [act 410]. The server 120 may filter the query log by language (e.g., English, French, Spanish, etc.), filter out misspelled words, filter out bad audio data, and/or filter out words that are not desirable.

[0044] The server 120 may then perform statistical analysis on the query log [act 415]. The server 120 may, for example, determine the most frequent queries, the most frequent words, the number of frequent words that cover a certain proportion of queries or query words, etc. The server 120 may also construct statistical language models 420 by counting the occurrence of words in certain contexts, smoothing the counts to obtain better probability estimates, and pruning the models to obtain a satisfactory

size/effectiveness tradeoff. Language models 420 may be constructed for different users or different user groups. For example, the server 120 may construct a language model 420 for a user group consisting of English speakers with German accents and a different language model for a user group consisting of English speakers with French accents. As illustrated in FIG. 4, the statistical analysis process also produces a vocabulary 425. The vocabulary 425 provides a list of words and word compounds (i.e., words that commonly occur together) to be used during the speech recognition process described below.

[0045] The server 120 may phonetically transcribe the words in the vocabulary 425 [act 430]. Here, the server 120 may associate one or more phonetic transcriptions with each word in the vocabulary 425. This phonetic transcription may be performed manually or automatically by the server 120. As a result of performing the phonetic transcription, the server 120 produces a phonetic dictionary 435. The phonetic dictionary 435 associates a list of words (and compounds) with possible pronunciations.

[0046] In response to receiving the query logs [act 405], the server 120 may also perform acoustic training by recording actual audio samples [act 440]. These audio samples may be used to train acoustic models 445 that will be later used to aid in the speech recognition process. The server 120 may then store the language models 420, phonetic dictionary 435, and acoustic models 445 in memory [act 450]. The server 120 may, for example, store the language models 420, phonetic dictionary 435, and acoustic models 445 locally at the server 120 (e.g., in memory 330) or externally from the server 120.

[0047] The server 120 may perform the processing described above a single time or at predetermined times. For example, the server 120 may update the language models 420, phonetic dictionary 435, and acoustic models 445 at predetermined time intervals (e.g., every hour) or as new query logs are created.

[0048] FIG. 5 illustrates an exemplary process, consistent with the present invention, for performing a search. While the foregoing acts are described as being performed by a server, it will be appreciated that a client device may alternatively perform some of the acts described below.

[0049] Processing may begin with a server, such as server 120, receiving a voice query [act 505]. The voice query may be received via the server's 120 input device 340 or over the network 140 via a separate device, such as a client device 110.

[0050] The server 120 may process the received voice query in a well-known manner to form a digital audio signal [act 510]. For example, the server 120 may perform analog-to-digital conversion to convert the audio signal to digital form and may break the digital audio signal into short windows (e.g., 10-20 ms frames). In an implementation consistent with the present invention, the server 120 may also determine which language model 420 is best suited for this voice query. For example, the server 120 may determine that a language model 420 directed to English speakers with German accents is best suited for this query.

[0051] The server 120 may then perform acoustic feature extraction in a well-known manner [act 515]. Within each of the short windows, the server 120 may look for

acoustic features to identify the sound that was spoken, derive a short feature vector, and classify the feature vector into a small number of categories.

[0052] The server 120 may perform speech recognition processing in a well-known manner on the feature vectors to derive word hypotheses [act 520]. The server 120 may analyze the feature vectors using the phonetic dictionary 435 that links one or more acoustic representations to words, the language model 420 to assign a probability value to different possible sequences of what could have been spoken, and acoustic models 445 to match the sequence of feature vectors with actual sound units. The speech recognition processing results in a list of the n-best word hypotheses and/or a word graph 525.

[0053] In an implementation consistent with the present invention, the server 120 may associate a weight with each possible word or word combination. The server 120 may determine these weights from confidence scores from the speech recognition processing, a priori probability from the language model, or, as will be described in more detail below, the number of documents resulting from a search or the frequency of the words/compounds in the resulting documents. Alternatively, the server 120 may use a combination of these techniques for determining weights.

[0054] Assume, for example, that a user wanted to search for information relating to the White House. Upon receiving the voice query, the server 120 may determine that the user query contained the following possible words "white," "light," "house," and "mouse." FIGS. 6A and 6B illustrate an exemplary n-best hypothesis list 600 and a word graph 650, respectively, that may be produced by the server 120. As illustrated in FIG. 6A, the n-

best hypothesis list 600 may contain a list of possible words or word-combinations that may be included in the voice query, along with associated weights. For example, the server 120 may determine that the voice query contains the word combinations "white house," "light house," "white mouse," or "light mouse" and may associate weights of 0.8, 0.73, 0.6, and 0.35, respectively, with these word combinations.

[0055] Alternatively, the server 120 may, as illustrated in FIG. 6B, produce a word graph 650 containing all word combination possibilities with associated weights. As illustrated, the server 120 may associate a weight of 0.8 with the word "white," a weight of 0.7 with the word "house," a weight of 0.5 with the word "light," and a weight of 0.4 with the word "mouse." As described above, the server 120 may determine these weights from confidence scores from the speech recognition processing, a priori probability from the language model, or search results.

[0056] The server 120 may set a group of query constraint parameters 530. These parameters may include the number of hypotheses to be considered (T), the total number of words to be included in a query (WordLimit), and the proportion of new words added from a first query possibility to a next query possibility (ProportionNewWords). These parameters may be automatically set by the server 120 or may be set manually. Moreover, these parameters may vary by user or user group.

[0057] Using the query constraint parameters 530 and the query term weights 532, the server 120 may construct a search query from the hypothesis list or word graph [act 535]. The server 120 may construct the search query to cover all (or the most likely) possible

hypotheses. FIG. 7 illustrates an exemplary process, consistent with the present invention, for constructing a search query. Assume in act 520 that the server 120 produces an n-best hypothesis list 525. Using the hypothesis list 525, the server 120 may determine the length (MinLen) of the shortest hypothesis within the top T hypotheses [act 710].

[0058] The server 120 may then remove noise words, such as "the," "of," "for," etc., that were incorrectly inserted by the server 120 during the speech recognition process to prune each hypothesis up to the length MinLen [act 720]. The server 120 may determine the length (MaxLen) of the longest pruned hypothesis [act 730]. The server 120 may, for example, determine MaxLen via a comparison operation. The server 120 may select k hypotheses from the n-best hypothesis list 525 [act 740], where

[0059] $$k = 1 + \frac{\text{WordLimit} - \text{MaxLen}}{\text{MaxLen} * \text{ProportionNewWords}}.$$

[0060] The server 120 may then obtain the weights 532 for the selected hypotheses [act 750]. The server 120 may form a weighted boolean query [act 760]. For the example above, the server 120 may produce the following boolean query:

[0061] 0.8(white house) OR 0.73(light house) OR 0.6(white mouse) OR 0.35(light mouse).

[0062] Alternatively, the server 120 may produce the following query:

[0063] $$\begin{pmatrix} 0.8\ \text{white} \\ \text{OR} \\ 0.5\ \text{light} \end{pmatrix} \text{AND} \begin{pmatrix} 0.7\ \text{house} \\ \text{OR} \\ 0.4\ \text{mouse} \end{pmatrix}.$$

[0064] In forming the boolean search query, terms may be repeated if necessary. For example, assume that the server 120 produces the following hypothesis list:

[0065] AB

[0066] CDE

[0067] FGH.

[0068] Assuming that each of these hypotheses is weighted equally, the server 120 may produce the following search query based on this hypothesis list:

[0069] $\begin{pmatrix} A \\ OR \\ C \\ OR \\ F \end{pmatrix}$ AND $\begin{pmatrix} B \\ OR \\ D \\ OR \\ G \end{pmatrix}$ AND $\begin{pmatrix} B \\ OR \\ E \\ OR \\ H \end{pmatrix}$.

[0070] Since the first hypothesis (AB) includes fewer terms than the other hypotheses, the server 120 may reuse one of the terms from the first hypothesis.

[0071] Once the weighted search query has been formed, the server 120 may, through the use of the search engine 125, perform a search using the query via any conventional technique [act 540]. The server 120 may then tailor the search results based on the query term weights 532. For example, the query term weights 532 may be provided as an input to the search engine 125 along with the query terms. The search engine 125 could use the query term weights 532 to determine how to rank the search results. For the example above, the search engine 125 may boost the ranking of a search result that contains "white house" compared to one that contains "light mouse," since "white house" is weighted

more heavily.

[0072] The server 120 may also use the query term weights to filter (or organize) the search results obtained by the search engine 125. For example, suppose the search engine 125 normally displays 10 search results per query. The server 120 may use the relative weights of the different hypotheses/terms to ensure that the first 10 results contain results that are proportional to the relative weights. As an example, the relative weight associated with "white house" in Fig. 6A is 0.32 (i.e., 0.8/2.48), and the relative weight of "light mouse" is 0.14 (0.35/2.48). Using these weights, the server 120 could filter the search results so that 3.2 (rounded to 3) of the first 10 search results are relate to "white house" and 1.4 (rounded to 1) of the results relate to "light mouse." Furthermore, it may be desirable to list the "white house" search results before the "light mouse" search results due to its higher relative weight. It will be appreciated that other ways of filtering search results using the query term weights may alternatively be used.

[0073] In another implementation consistent with the present invention, the server 120 may use the query term weights to eliminate (i.e., not use as part of the search query) hypotheses or terms that have a weight/confidence score below a predefined threshold value. For example, assume that the threshold was set such that hypotheses with a weight 0.4 or below should be eliminated. For the hypothesis list provided above with respect to FIG. 6A, the server 120 may eliminate the hypothesis "light mouse" since it is associated with a weight below the threshold value of 0.4.

[0074] Once the search results have been obtained, the server 120 may use these

search results to refine the search query. For example, assume that the server 120 constructs a boolean search query using the hypotheses listed in FIG. 6A. Assume further that none of the documents obtained by the search engine 125 correspond to the hypothesis "light mouse." In such a case, the server 120 may discard that hypothesis from the original list, create a new boolean search query using the remaining hypotheses, and then perform a search using the new search query. The server 120 may perform this iteration once, or repeatedly until, for example, each of the hypotheses has search results associated with it.

[0075] As an alternative to deleting a hypothesis if there are no corresponding search results, the server 120 may modify the weights (confidence scores) of those hypotheses based on the contents of the documents corresponding to the search results. Here, the server 120 may, for example, increase the weights associated with those hypotheses or terms relating to a high number of results.

[0076] In yet a further implementation consistent with the present invention, the server 120 may consider compounds in performing or refining the search. In some conventional searching techniques, a search engine may obtain better results for the compound "new york restaurants" than the search engine would return if the terms "new," "york," and "restaurants" were separately entered. According to this exemplary implementation, the server 120 may develop the n-best hypothesis list 525, feed it into a search engine 125, evaluate the search results to identify compounds, and revise the hypothesis list based on the identified compounds. As an alternative to identifying

compounds after receiving search results, the server 120 may detect compounds prior to constructing the search query 535. In such a situation, the server 120 may then replace an existing hypothesis with the compound when constructing the search query.

[0077] Once the search results have been obtained, the server 120 may then provide the results to the user via the client 110 [act 545]. The server 120 may, for example, cause the results to be displayed to the user.

[0078] CONCLUSION

[0079] A system and method consistent with the present invention provide a voice interface for search engines. Through the use of a language model, phonetic dictionary, and acoustic models, a server generates an n-best hypothesis list or word graph. The server uses the n-best hypothesis list or word graph to construct a search query to cover possible possibilities. As a result, the server is capable of returning relevant search results for even queries containing few words.

[0080] The foregoing description of exemplary embodiments of the present invention provides illustration and description, but is not intended to be exhaustive or to limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practice of the invention. For example, while series of acts have been presented with respect to FIGS. 5 and 7, the order of the acts may be altered in other implementations consistent with the present invention. No element, act, or instruction used in the description of the present application should be construed as critical or essential to the invention unless explicitly described as such.

[0081] The scope of the invention is defined by the following claims and their equivalents.